基于IPv6的UDP通信的实现

石炎生,刘利强
(湖南理工学院 计算机系,湖南 岳阳 414006)

摘 要:IPv6是为了克服现在IPv4协议的缺陷而设计的下一代因特网协议。随着Internet的飞速发展,IPv6(Internet Protocol Version 6)必然会取代IPv4,开发支持IPv6的网络应用程序已经迫在眉睫。文中首先对IPv6进行了简介;然后介绍了UDP的通信原理、IPv6下Socket地址结构和IPv6下Socket函数,并给出了在IPv6下基于无连接的Socket的客户机/服务器模型和利用UDP协议进行网络通信的方法;最后实现了在IPv6协议下服务器端和客户端之间数据信息的传输,相信对今后开发基于IPv6的大型网络应用程序会有极大的帮助。

关键词:IPv6;UDP;Socket;网络编程
中图分类号:TP393 **文献标识码**:A **文章编号**:1673-629X(2006)05-0191-03

Realization of UDP Communication Based on IPv6

SHI Yan-sheng, LIU Li-qiang
(Department of Computer, Hunan Institute of Science and Technology, Yueyang 414006, China)

Abstract: IPv6 is intended to be the next network layer protocol of the Internet and it is designed to overcome the limitations of the current IPv4 protocol. With the quick development of Internet, the IPv6(Internet protocol version 6) certainly can replace IPv4, developing and supporting network applications program of IPv6 extremely urgent. This article introduces IPv6, UDP communication principle, Socket address structure and Socket function, and provides the client/server model of Socket on non-connect and method of network communication on UDP protocol for IPv6. Believe developing based on IPv6 in the large network applications program can have great help.

Key words: IPv6; UDP; Socket; network programming

0 引 言

随着计算机技术的发展和网络的不断普及,目前的IPv4协议在许多方面已经显得不太适应,如IPv4地址严重不足、缺乏服务质量控制QoS、安全性差和移动接入不方便等问题。IETF(Internet Engineer Task Force)组织已意识到:目前的IPv4网络协议已很不适用了,因此提出了下一代的网络协议IPv6。IPv6是为适应未来对于网络基础设施的数量和质量的需求而设计的下一代互联网协议,许多机构都已经开始了这方面的测试和研究,并且逐渐被主流的操作系统和网络硬件设备所支持。IPv6协议作为目前热门的研究领域,相应的协议和标准还在不断的制订、讨论和变化中,这些都需要人们积极地参与研究、开发和实践。文中将探讨基于IPv6的UDP网络通信程序设计方法。

1 UDP的通信原理

UDP(User Datagram Protocol,用户数据包协议)是一种无连接的通讯协议。UDP协议将独立的数据包从一台计算机传输到另外一台计算机,但是并不保证接受方能够接收到该数据包,也不保证接收方所接收到的数据和发送方所发送的数据在内容和顺序上是完全一致的。因此,UDP协议更类似于普通邮政服务,寄信人不能够保证所寄出去的信能够被收信人及时收到,后发出的信也许会比先发出的信更早到达。由于UDP协议比较简单,UDP头包含很少的字节,比TCP负载消耗少,适用于对可靠性要求不高的应用环境。图1是使用无连接Socket的客户机/服务器模型。

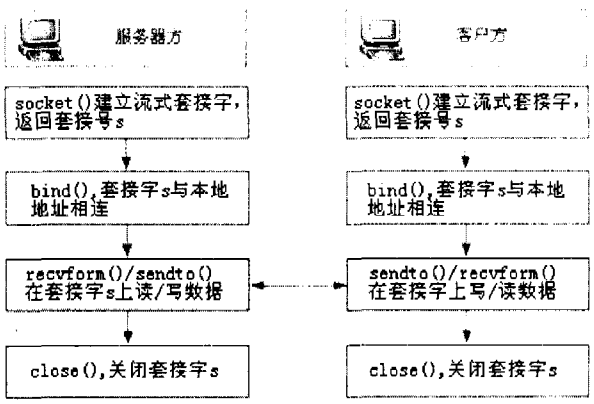

图1 基于无连接的Socket的客户机/服务器模型

收稿日期:2005-11-17
基金项目:湖南省自然科学基金项目(05JJ40110)
作者简介:石炎生(1961-),男,湖南岳阳人,副教授,研究方向为计算机网络及虚拟现实技术。

2 IPv6 下 Socket 套接字

2.1 IPv6 的 Socket 地址结构

IPv6 使用 128 位地址,定义了本身的专用 Socket 地址结构[1]。

(1) sockaddr_in6 结构。

```
struct sockaddr_in6 {
sa_family_t sin6_family; //必须是 AF_INET6
in_port_t sin6_port; //传输层端口号
uint32_t sin6_flowinfo; //IPv6 拥塞级别与流信息
struct in6_addr sin6_addr; //128 位的 IPv6 地址
uint32_t sin6_scope_id; //IPv6 的接口范围
};
```

其中,sin_family 表示地址所属的协议族,对于 IPv6 地址来说,始终为 AF_INET6;sin6_flowinfo 包含 IPv6 流信息;sin6_addr 标识一个 128 位的 IPv6 地址,in_addr6 结构如下所示:

```
struct in6_addr{
uint8_t s6_addr[16]; //IPv6 地址
};
```

(2) addrinfo 结构。

```
struct addrinfo {
int ai_flags; //AI_PASSIVE, AI_CANONNAME,AI_NUMERICHOST
int ai_family; //必须是 AF_INET6
int ai_socktype; //字节流用 SOCK_STREAM,数据报用 SOCK_DGRAM
int ai_protocol; //TCP 协议用 IPPROTO_TCP,UDP 协议用 IPPROTO_UDP
socklen_t ai_addrlen; //ai_addr 地址长度
char * ai_canonname; // 规范名
struct sockaddr * ai_addr; // sockaddr 地址结构
struct addrinfo * ai_next; // 指向下一个 addrinfo 结构
};
```

2.2 IPv6 的 Socket 函数

IPv6 的 Socket API 函数中一部分沿用了 IPv4 的 Socket 函数,也新增了一些 IPv6 专用的 Socket 函数,IPv6 的 Socket 函数如表 1 所示[2]。

1) getaddrinfo() 函数原形为:int getaddrinfo (const char * hostname, const char * servname, const struct addrinfo * hints, struct addrinfo * * res)

其中 hostname 可以是主机名地址或者是 IPv6 的地址;service 可以是服务名或十进制端口号,这给编程也带来了很大的方便;Hints 相当于一个过滤器,只有符合 Hints 结构的内容才会返回到 res 指针中。

2)socket() 函数原形为:int socket (int domain, int type, int protocol)

其中第一个参数确定协议族,如符号常数 AF_INET6 表示用 IPV6 协议;第二个参数指明了想使用的通信服务类型,如符号常数 SOCK_DGRAM 表示数据报,用 SOCK_STREAM 表示字节流;第三个参数指明此 Socket 请求使用的协议,如符号常数 IPPROTO_TCP 表示用 TCP 协议,用 IPPROTO_UDP 表示用 UDP 协议[3]。

表 1 IPv6 的 Socket 函数

IPv6 函数	功能说明
inet_ntop()	字符串地址转为 IP 地址
inet_pton()	IP 地址转为字符串地址
getipnodebyname()	由名字获得 IP 地址
getaddrinfo()	获得全部地址信息
getnameinfo()	获得全部名字信息
socket()	建立 Socket
bind()	Socket 与地址绑定
listen()	网络监听
accept()	接收 TCP 连接
connect()	建立 TCP 连接
send()	发送数据(TCP)
sendto()	发送数据(UDP)
recv()	接收数据(TCP)
recvfrom()	接收数据(UDP)
close()	关闭连接

3)listen()函数的函数原型为:int listen(int sockfd, int backlog)

其中 sockfd 是 Socket 函数返回的 Socket 描述符;backlog 指定在请求队列中允许的最大请求数。

4)sendto()函数原型为:int sendto(int sockfd, const void * buf, int len, unsigned int flags, struct sockaddr * ai_addr, socklen_t ai_addrlen);

其中 sockfd 是你想用来传输数据的 Socket 描述符;buf 是存放发送数据的缓冲区;len 是以字节为单位的数据的长度;flags 一般情况下置为 0,struct sockaddr * ai_addr 确定地址,socklen_t ai_addrlen 指明目的地址的大小。

5)recvfrom()函数原型为:int recvfrom(int sockfd, void * buf, int len, unsigned int flags, struct sockaddr * addr, int * addrlen);

其中 sockfd 是接受数据的 Socket 描述符;buf 是存放接收数据的缓冲区;len 是缓冲的长度;flags 也被置为 0,addr 通常是一个指向 sockaddr 变量的指针;addrten 通常为一个指向值为 sizeof(struct sockaddr)的整型指针变量[4]。

3 编程实现

3.1 服务器端程序设计

服务器端首先启动,通过调用 Socket()建立一个 Socket,然后调用 bind()将该 Socket 和本地网络地址联系在一起,接着服务器端就可以通过 recvfrom()和 sendto()来接收和发送数据。最后,待数据传送结束后,调用 close

()关闭 Socket[5]。服务器端设计过程如下。

(1) 创建服务器端套接字。

```
memset (&hints, 0, sizeof (hints));
hints.ai_family = AF_INET6; //指定用 IPV6 协议
hints.ai_socktype = SOCK_DGRAM; //指定用数据报
hints.ai_protocol = IPPROTO_UDP; //指定用 UDP 协议
hints.ai_flags = AI_NUMERICHOST; //IP 用数字表示
rc = getaddrinfo ("::1", "5001", &hints,&res); //解析本机地址
s_send = socket (res->ai_family, res->ai_socktype, res->ai_protocol);//创建 socket
```

(2) 绑定本机监听端口。

```
rc=bind(s_send,res->ai_addr,res->ai_addrlen);
```

(3)接收数据。

```
recvfrom( s_ send, buf, sizeof ( buf ), 0, ( struct sockaddr * ) &sin,&sin_len)); //接受数据到 buf
printf("recvfrom:%s",buf); //打印接受的数据
```

(4) 关闭套接字。

```
close(s_send );
```

3.2 客户端程序设计

客户端创建一个 Socket,然后调用 bind()将该 Socket 和远地网络地址联系在一起,客户端就可以通过 recvfrom ()和 sendto()来接收和发送数据。最后,待数据传送结束后,调用 close()关闭 Socket。客户端设计过程如下。

(1) 创建客户端套接字。

```
memset (&hints, 0, sizeof (hints));
hints.ai_family = AF_INET6; //指定用 IPV6 协议
hints.ai_socktype = SOCK_DGRAM; //指定用数据报
hints.ai_protocol = IPPROTO_UDP; //指定用 UDP 协议
hints.ai_flags = AI_NUMERICHOST; //IP 用数字表示
rc = getaddrinfo ("3ffe:3211::1", "5001", &hints,&res);
//解析服务器地址
s_send = socket (res->ai_family, res->ai_socktype, res->ai_protocol); //创建 socket
```

(2)绑定远地监听端口。

```
bind (s_send,res->ai_addr,res->ai_addrlen);
```

(3) 发送数据。

```
sendto(s_send,buf,sizeof(buf),0,res->ai_addr,res->ai_addrlen);
```

(4) 关闭套接字。

```
close (s_send);
```

4 结束语

随着 Internet 技术的不断发展,IPv6 作为一种新的 Internet 协议必定会取代 IPv4 成为下一代的互联网协议,开发支持 IPv6 的网络应用程序的问题会变得越来越重要。开发在 IPv6 下用 UDP 协议的服务器和客户机应用程序,能够较好地实现服务器端和客户端之间数据信息的传输,对开发基于 IPv6 的大型网络应用程序会有极大的帮助。

参考文献:

[1] RFC2553. Basic Socket Interface Extensions for IPv6[S]. 1999.

[2] RFC2292. Advanced Sockets API for IPv6[S]. 1998.

[3] RFC3493. Basic Socket Interface Extensions for IPv6[S]. 2003.

[4] Davies J. 理解 IPv6[M]. 张晓彤等译. 北京:清华大学出版社,2004.

[5] Stevens W R. TCP/IP 详解(卷 3)[M]. 胡谷雨等译. 北京:机械工业出版社,2000.

(上接第 190 页)

据。

(2) 技术特色:

a.采用先进的分布式多层架构,应用了 COM +, VS.NET 等领先技术。

b.解决了远程大型数据库数据共享的难题。

c.基于 Web 的数据库通用查询和对比分析。

d.基于 Web 的报表定制技术。

e.提供了报表、曲线、图形等丰富的表现形式。

7 结束语

永宁钻采公司管理信息系统的建立,可以充分利用各种"信息资源为生产服务;实现整个公司数据共享"[3],提高了永宁钻采公司的生产管理透明度,使相关工序、相关部门可随时了解彼此间生产状况,及时进行生产调整。有关人员可以从繁重的数据处理、统计计算、报表加工工作中解放出来,多花一些时间去考虑生产中存在的问题,多做一些生产分析、质量分析,去寻找提高产品产量和产品质量的技术改进方法和管理措施。"系统的建立,有助于带动和提高管理人员的素质[4]。"可以肯定地讲,系统的建立将给生产管理带来可观的社会效益和经济效益,同时也将永宁钻采公司的生产管理推上一个新台阶。

参考文献:

[1] 刘　渝,李　涛,蒋红红. 油田生产运行信息管理系统[J]. 油气田地面工程,1999,22(7):13-14.

[2] 张培宏,陈武新. 油田生产实时监控系统设计方案探讨[J]. 测绘通报,2004(5):26-28.

[3] 汤　军. 采油厂静态信息 GIS 综合管理系统设计与实现[J]. 江汉石油学报,2004,26(3):33-34.

[4] 陈　虎,孙　鹏. 地理信息系统技术在油田开发方案设计中的应用[J]. 电脑开发与应用,2000,16(9):29-31.